

研 究 報 告 書

「柔軟性と電力効率に優れた次世代専用プロセッサ設計手法の開発」

研究期間：2018年10月～2020年3月

研究者番号：50177

研究者：谷本 輝夫

1. 研究のねらい

多様な組み込み機器により物理世界の情報取得、操作を行うIoT (Internet of Things)の普及が進んでいる。これに伴い、コンピュータ・システムへの要求に以下の大きな変化が見込まれる。

- ・ 機器多様化による短期間開発、多品種少量生産

物理世界と相互作用するためには具体的な応用ごとの設計が必要であり、多機能化ではなく多様化は避けられない。現在数年を要するハードウェアの設計、検証、そして生産に要する期間を短縮し、少品種大量生産から多品種小量生産へ転換する必要がある。

- ・ 極めて高い電力当たり性能(電力効率)の実現

IoT 機器は情報取得後、操作量等の計算に性能制約を持つ。また、体内埋め込み型や装着型など電力供給に小容量電池を用いるものが多い。このような極めて高い電力効率要求に応えるため、汎用プロセッサではなくアプリケーション特化プロセッサ中心の構成が必要である。

- ・ 出荷後システム再構成の柔軟性

機器間通信によるシステムである以上、新旧機器混在時の通信担保や、セキュリティパッチ適用など自己再構成する必要がある、ある程度の汎用処理能力が必要である。

これらに対応するため、最大公約的ハードウェア上にソフトウェアを実装する構成から、柔軟性を維持しつつ専用ハードウェアを中心とした構成へ設計思想を転換する必要がある。そこで、本研究は柔軟性と電力効率に優れた専用プロセッサを短期間開発する次世代設計手法の確立を目的とし、汎用／専用ハイブリッドプロセッサ自動設計手法の確立を目指す。

汎用／専用ハイブリッドプロセッサとは、汎用処理回路と専用処理回路を密結合した単一プロセッサであり、回路面積に応じたアプリケーションの専用回路実装と細粒度専用処理を両立できる。これまで取り組んできた汎用プロセッサにおけるマイクロアーキテクチャレベルでの命令実行のクリティカルパス解析ツール開発の知見を活かしてプログラム中の専用化有効部分の抽出と、優先度評価に基づく自動専用プロセッサ設計を目指す。

この技術により、現在プログラマがソフトウェアを作成するのと同様に、プログラムを専用ハードウェア化しすぐに所望の専用プロセッサを得られる設計自動化ツールが必要である。これを世界に先駆けて開発し、従来のプロセッサ設計の在り方の変革を目指す。

2. 研究成果

(1) 概要

汎用／専用ハイブリッドプロセッサ設計手法の確立に向けて本研究では大きく3点について取り組んだ。1点目は汎用プロセッサ上でプログラムを実行した際の専用化有効部分の抽出であり、2点目は DSL (Domain-Specific Language) を利用した専用回路実装における資源配分の最適化手法の確立、3点目は実アプリケーションを対象としたケーススタディの実施である。

1点目の専用化有効部分の抽出については、アウトオブオーダー命令実行の依存グラフ表現におけるクリティカルパス解析を発展させ、プログラム中における専用化によって高速化が期待できる処理部分の抽出が可能となった。これらの処理部分を命令チェーンのまとまりとしてとらえ、その処理を高速化した際に得られる性能向上の最大値をサイクル数単位で得ることができる。この解析は、一度プロセッサシミュレータ上でプログラムを実行しプロファイルを取得した後、DAG (Directed Acyclic Graph) を構築し、そのグラフ上で解析を行うことで実施できる。これは、複数回のシミュレーションにより同様の結果を得る手法に比べて非常に効率的に解析可能である。

2点目の DSL を利用した専用回路実装における資源配分の最適化手法に関しては、専用化設計の自動化を考える上ではそのプログラミングモデルから考えることが不可欠であるという考えに至り、画像処理はテンソル計算に特化した DSL である Halide に着目し、これを用いたハードウェア生成における回路資源の最適化手法に取り組むことにした。一般にアプリケーションは複数の独立した処理(関数)によって構成され、それらを同時に専用回路としてハードウェア実装する際の資源配分を考える。それぞれの処理は回路量と処理時間にトレードオフを持ち、回路生成時のパラメタ設定によりそのトレードオフを調節することが可能である。本研究では、最適化を処理内におけるパラメタ探索と、複数処理を組み合わせた時のパラメタ探索の2段階に分けることで探索時間の短縮を図った。

3点目として、対象アプリケーションとして自動運転を始めとした自律動作する機器で用いられる SLAM (Simultaneous Localization and Mapping) を取り上げ、実際に設計フローを適用してアプリケーションのハードウェア・ソフトウェアハイブリッド実装を行った。結果として、ATAM (Abecedary Tracking and Mapping) のフレームレートを 4.88 倍に向上できることが分かった。

(2) 詳細

テーマ1「専用化有効部分の抽出」

本テーマではプログラム中の処理の中で専用化可能な命令チェーンを抽出することを目的に抽出方法の確立を目指して取り組んだ。専用化方法としては、数～数十命令の命令列を専用命令として実装することを想定した。これが可能になると、定性的には1) 命令数削減によるフロントエンドの処理軽減、2) 専用化された命令列ないでのレジスタファイル読み書きが不要、3) ILP (Instruction Level Parallelism) を専用命令として回路実装することで演算に要する遅延の削減等の効果が期待できる。

本研究では、動的命令中の N 入力1出力となる命令集合を依存グラフ上で探索するアルゴリズムを考案・実装し、SPEC CPU2006 ベンチマークにおいて専用命令化により性能向上可能なポテンシャル評価を行った。本手法の一部は論文 [1-1] として出版された。図1に1万命令を対象に N=3 と仮定して解析した結果を示す。N=3 は x86 などの命令セットでも採用されている命令フォーマットで実装可能であり、実現可能性の高い状況を想定している。青および橙で示す実行時間内訳は専用化による性能向上が見込めないとされる割合を示しており、グレーで示す領域が専用化により高速化可能な実行時間の割合を示している。Gromacs や gobmk などのアプリケーションでは 30%程度の潜在的な高速化の余地があることが分かり、専用命令化による高速化は有力なアプローチであると言える。一方で、mcf や cactusADM は N=3 の専用命令化では高性能化する余地はほとんどないことが分かる。この結果のように、ハードウェア設計の前に依存グラフを用いた高速な解析により潜在的な効果を評価可能であることは重要な意義を持つ。

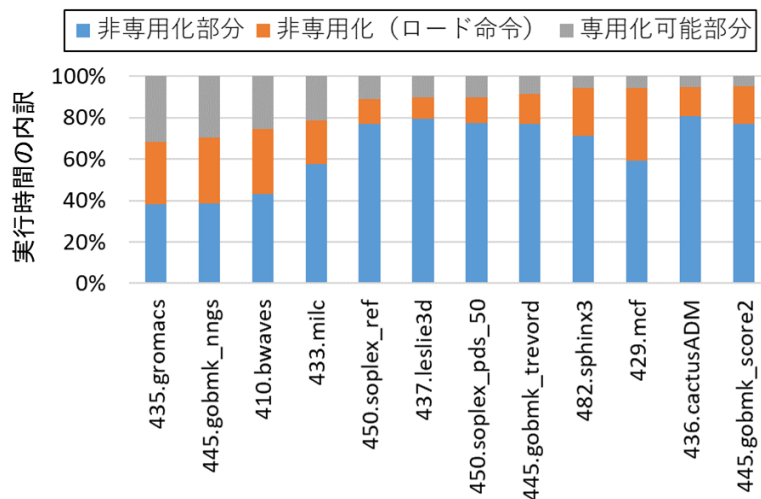


図1 SPEC CPU2006 における 3 入力 1 出力専用命令化の潜在的効果の評価結果

テーマ2「DSL を利用した専用回路実装における資源配分の最適化手法」

本テーマでは、汎用／専用ハイブリッドプロセッサの自動設計を実現するためにはプログラミングモデルが重要であるとの考えに基づき、DSL を用いたプログラム記述からハードウェアを生成する手法に関する研究を行った。処理を専用回路化する際には、プログラム本体との入出力を定義しそのインターフェースを作成する必要がある。その際には、副作用を持たない関数に着目するのが良い。純粋関数型言語では、関数の入出力が明確に定義されており、副作用を持たないことが言語仕様で担保されているため関数を専用化の単位とみなすのに都合が良い。また、関数型言語では、データフローの解析が容易なため、それをハードウェアのパイプラインとして実装するのに適している。

このようなことから、本研究では Halide という DSL に着目し、その FPGA バックエンドを開発している Fixstars 社の協力の元、アプリケーション実装を想定したリソース最適化手法の確立を目指した。Halide はプログラムの記述がアルゴリズム部とスケジューリング部に分かれ

ている特徴を持つ。アルゴリズム部は演算内容を規定し、スケジューリング部は処理するデータの範囲(量)や順序、並列度などを規定する。

Halide はもともと CPU や GPU などといった汎用的なプロセッサで実行することを想定して作成された言語であるが、スケジューリング部のパラメタを自動で探索することで性能を最適化できることが知られている。この特性をハードウェア化(FPGA 化)の際にも活用できると考え、あらかじめ設定した設計空間(処理・データ転送の並列度)に対して関数(モジュール)ごとの回路資源量と性能のトレードオフ探索と、複数の関数(モジュール)を同時に FPGA に実装する際の回路資源配分の探索を2段階で行う方法を確認した(図2)。これらの探索は並列かつ自動的に実行可能である。

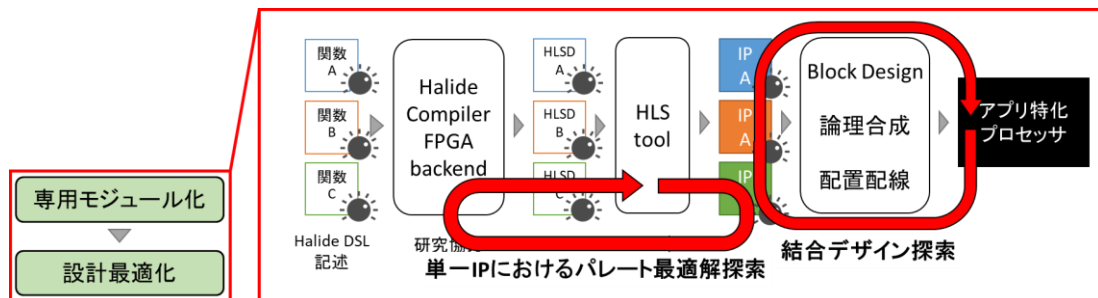


図2 単一 IP におけるパレート最適解探索と総合デザイン探索の2段階探索方法

テーマ3「実アプリケーションを対象としたケーススタディ」

対象アプリケーションとして SLAM の実装である ATAM を取り上げ、テーマ2で考案した最適化手法を用いてハードウェア・ソフトウェアハイブリッドシステムとして実装した[3-1][3-2]。アプリケーション中の主な処理として、特徴量検出、特徴量計算、特徴量マッチング、特徴点追跡の4つの画像処理を専用化対象として選定し、それらを Halide DSL で実装した。今回の実装では特徴量検出はカメラ入力のフレームレートを満たせばよいため、それ以外の3機能が回路資源配分最適化の対象である。

図3にモジュールごとの設計空間(パラメタ)探索の結果を示す。各モジュールに設定された設計空間に対し、縦軸に Lookup table の使用量、横軸に実行時間を示す。黒点は設計空間のうちパレート最適であるもので、これらを結合デザインで用いるパラメタの候補とする。

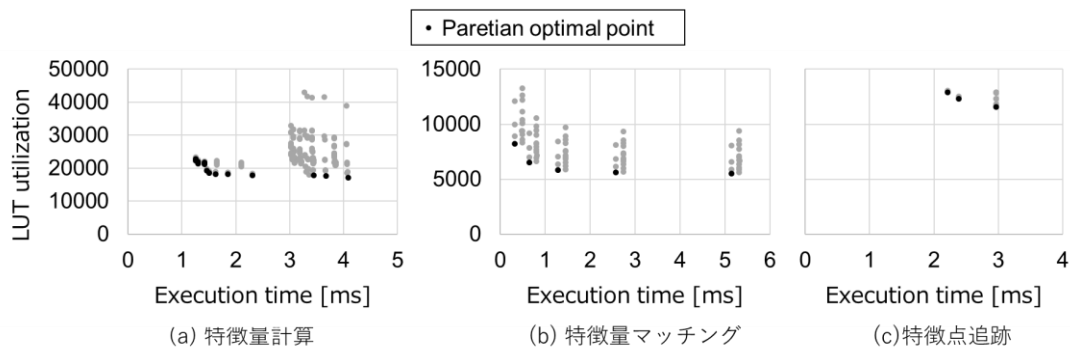


図3 モジュールごとの設計空間(パラメタ)探索の結果

結合デザインの最適化では、各モジュールのパレート最適なパラメタの組み合わせを探索することで最適な組み合わせを見つける。探索の結果、ATAM のフレームワークをソフトウェア実装と比較して最大で 4.88 倍向上することができた。

3. 今後の展開

Intel 社の One API や、Xilinx 社の Vitis など、プログラムを元に FPGA 実装を生成するアプローチが製品化されてきており、この流れはこれからますます成熟し利用が広まっていくと考えられる。本研究の成果は1)ツールとしての提供、2)本ツールを用いた専用実装サービスの提供などとして実用することができる。今後も研究協力先との連携を継続し、実用に向けて展開していく予定である。

4. 自己評価

限られた研究期間で設計手法を開発するという挑戦的なテーマに対し、実アプリケーションを対象に開発フローを実際に通すことができた。ポストムーア時代にはアプリケーションを専用実装し、その性能を最大限引き出す設計が不可欠であり、その設計手法の確立に寄与することができたと考えている。一方で、テーマ1の専用命令化については、専用化可能部分の抽出と専用化時のポテンシャル評価にとどまり、より現実的な性能評価や回路設計までには至らなかった。より実用に近い技術の確立を優先した結果ではあるが、専用命令化についても今後実装を進めていきたい。

設計空間探索は並列実行可能であり、本研究において複数台の計算機を用いて探索を行った。本課題の支援があればこそ、このような実験が可能であり、研究費により本研究が加速したと考えている。

アプリケーションごとのカスタムハードウェア実装は FPGA の高性能、大規模化に伴ってニーズが増加しており、本研究は利用可能なリソースを最大限性能に還元するための技術として有用である。今後その適用範囲を広げることでさらに高度なアプリケーションによるサービス創出につながると考えられる。

5. 主な研究成果リスト

(1) 論文(原著論文)発表

1. Teruo Tanimoto, Takatsugu Ono, and Koji Inoue, "Critical Path based Microarchitectural Bottleneck Analysis for Out-of-Order Execution," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E102-A, No.6, pp.758-766. Jun. 2019.

(2) 特許出願

研究期間累積件数: 0件

(3) その他の成果(主要な学会発表、受賞、著作物、プレスリリース等)

1. 原凌司, 井上優良, 谷本輝夫, 大澤隆志, 丸岡晃, 飯塚拓郎, 井上 弘士, 「高位合成用 DSL コンパイラを用いたコーナー検出処理のハードウェア実装」, 情報処理学会研究報告, Vol.2018-ARC-233 No.11, pp.1-8, 2018 年 11 月.
2. 原凌司, 谷本輝夫, 井上優良, 大澤隆志, 丸岡晃, 飯塚拓郎, 追川修一, 井上弘士, 「高位合成用 DSL コンパイラを用いた SLAM アプリケーションのハードウェアアクセラレーション」, 情報処理学会研究報告, Vol.2019-ARC-238 No.8, pp.1-10, 2019 年 11 月.